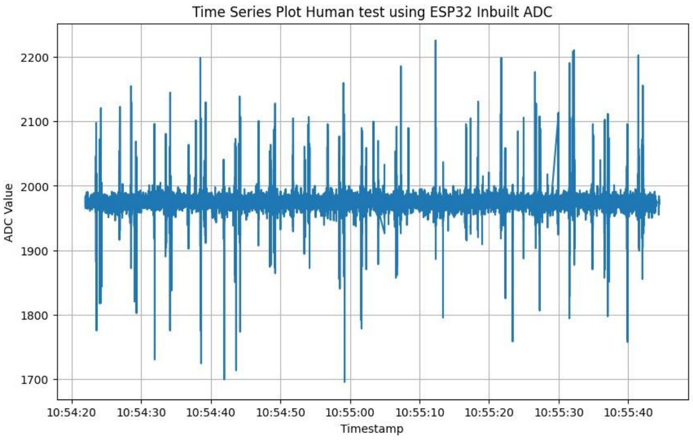
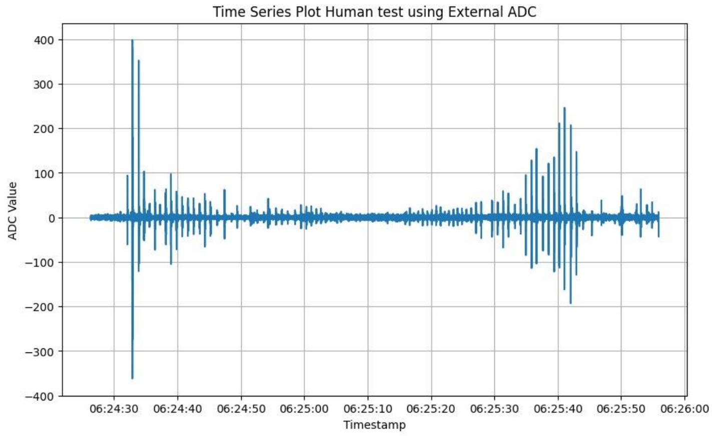
Comparison Between ESP32 inbuilt 12-bit ADC and the External 16-bit ADC (ADS1115)

Test -1 Human testing performed on both the apparatus





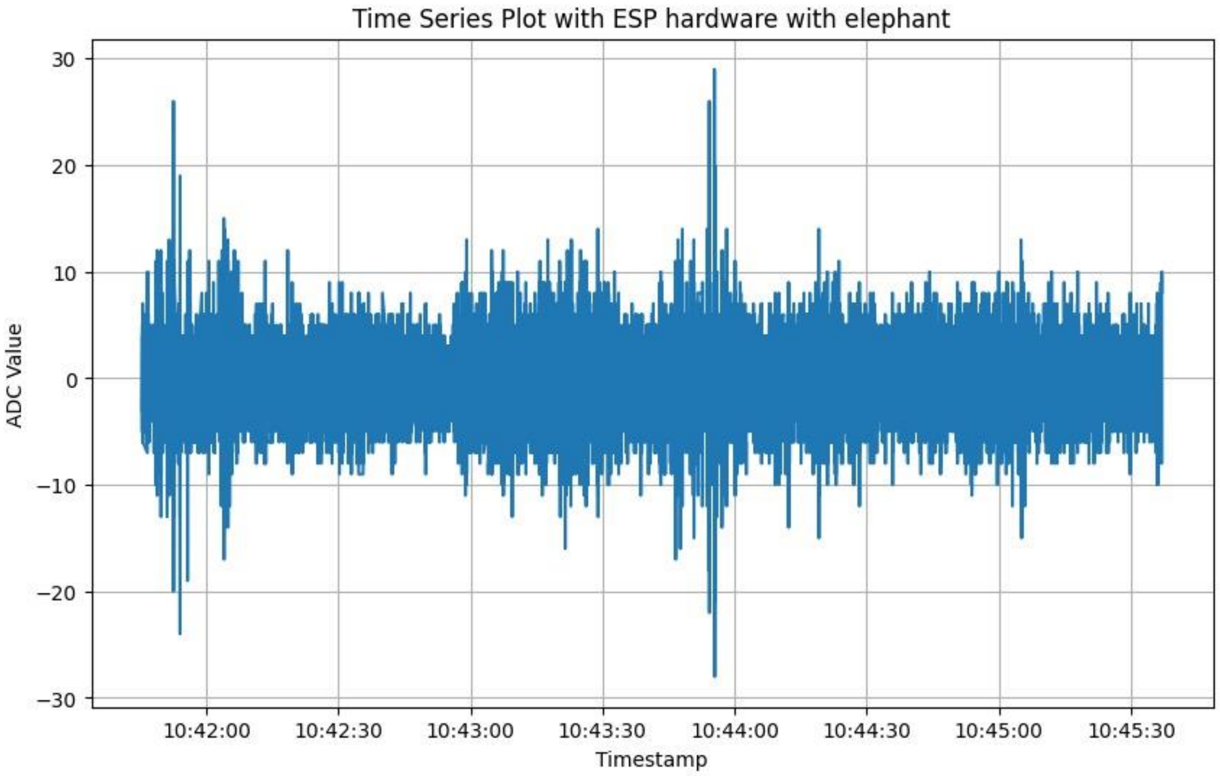
- The left plot represents the internal ADC and the right plot represents the external 16bit ADC we used on our esp32. we can clearly see in the two plots the left-hand side plot is noisy than the right-hand side plot.
- The pattern of the data seems to be clear when using the 16 Bit ADC rather than the inbuilt 12-bit ADC in the ESP32 microcontroller.
- The python Code snippet used to plot the Data is given below

```
import pandas as pd
import matplotlib.pyplot as plt

file_path = r'C:\Users\Sureka Siriwardana\Desktop\FYP\DATA\ZOO_Day_3\ESP_Inbuilt\data_2024-07-06_10-54-22.csv'
df = pd.read_csv(file_path)

print(df.columns)


print(df.head())


date_column = 'Timestamp'
value_column = 'ADC Value'

df[date_column] = pd.to_datetime(df[date_column], format='%H:%M:%S.%f')

df.set_index(date_column, inplace=True)


plt.figure(figsize=(10, 6))
plt.plot(df[value_column])
plt.title('Time Series Plot Human test using ESP32 Inbuilt ADC')
plt.xlabel('Timestamp')
plt.ylabel('ADC Value')
plt.grid(True)
plt.show()
```

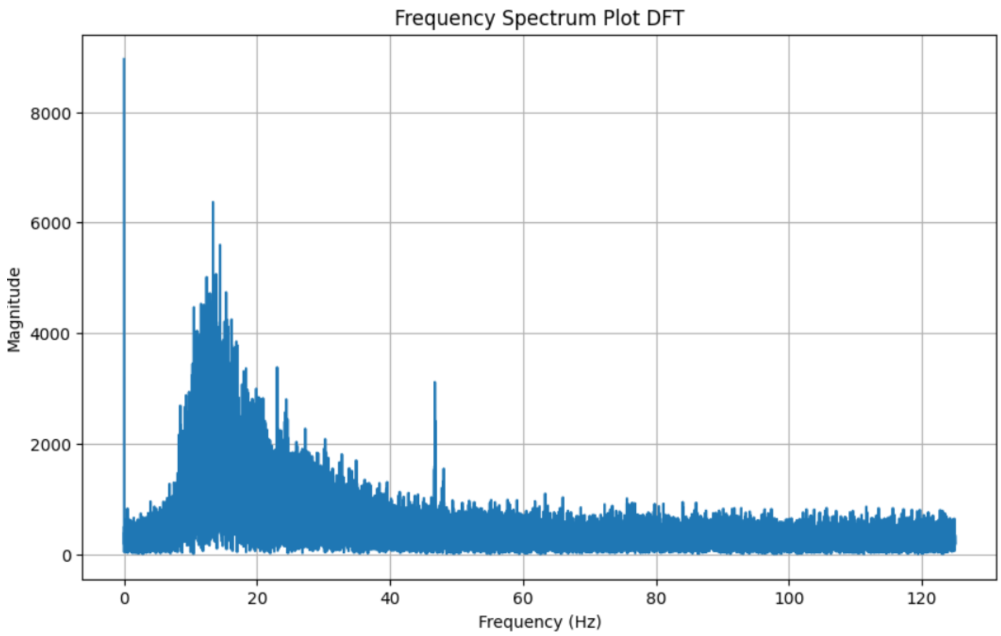
- After the above test were performed a elephant was used to walk towards us and go back in a distance of 50. Two tests were performed. Given below is the comparison of the Data collected using external 16 bit ADC vs ESP32 internal 12 bit ADC

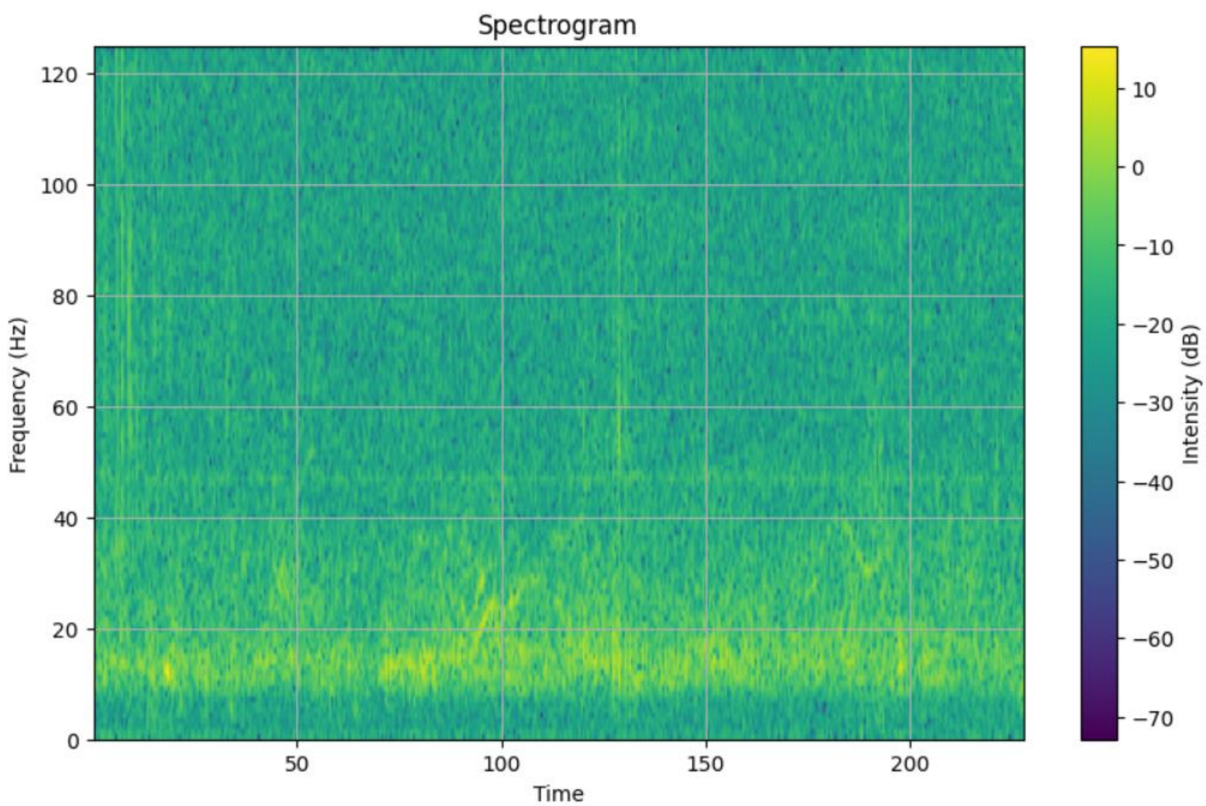
Test 2 – Using and Elephant to walk towards us and walk backwards again

1. Using external 16-bit ADC



- The Plot Shows the data we collected using the external 16-bit ADC.
- The Two plots given below shows the above plot in Fourier domain and a spectrogram analysis of the data is given below.
- In the above time series plot its observable that there is a frequency spectrum of the elephant footfall.



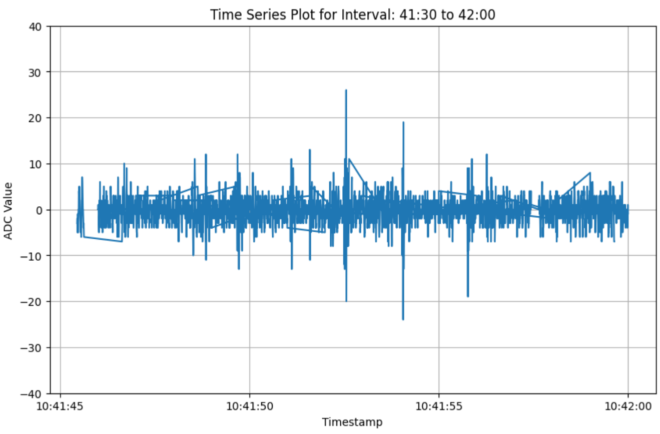


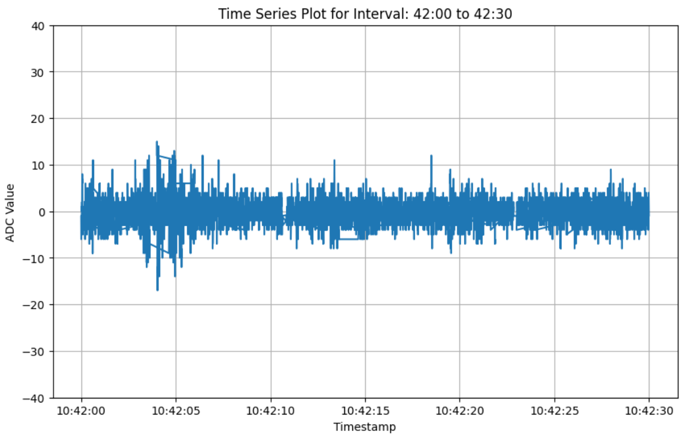
- The DFT plot and the spectrogram analysis clearly shows that the dominant frequencies of the footfall is observed between 0-50Hz frequency range.
- It’s clearly visible in the spectrogram in yellow colour there are frequencies in between 20 and 40.
- The python code used to plot these data and obtain the DFT plot and the spectrogram is is Given Below.

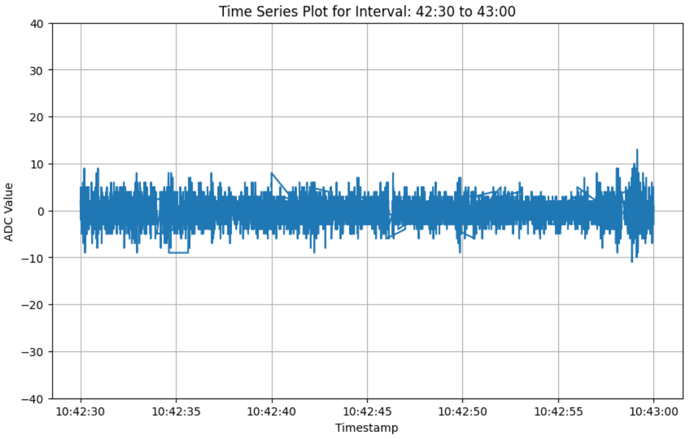
```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

file_path = r'C:\Users\Sureka Siriwardana\Desktop\FYP\DATA\ZOO_Day_4\ESP 16 bit External\Test 1 ESP front and back 16 bit.csv'

df = pd.read_csv(file_path)

print(df.columns)
print(df.head())

date_column = 'Timestamp'
value_column = 'Data'

df[date_column] = pd.to_datetime(df[date_column], format='%H:%M:%S.%f')

df.set_index(date_column, inplace=True)

plt.figure(figsize=(10, 6))
plt.plot(df[value_column])
plt.title('Time Series Plot with ESP hardware with elephant')
plt.xlabel('Timestamp')
plt.ylabel('ADC Value')
plt.grid(True)
plt.show()

adc_values = df[value_column].values
N = len(adc_values)
dft = np.fft.fft(adc_values)
dft_freq = np.fft.fftfreq(N, d=(df.index[1] - df.index[0]).total_seconds())
```

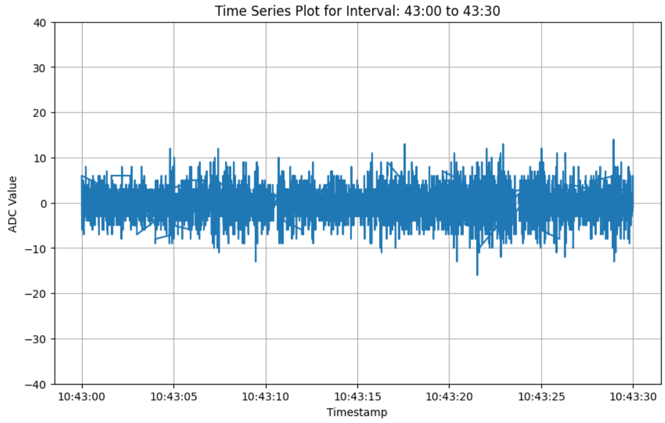
```python
plt.figure(figsize=(10, 6))
plt.plot(dft_freq[:N//2], np.abs(dft[:N//2]))
plt.title('Frequency Spectrum Plot DFT')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.grid(True)
plt.show()
plt.figure(figsize=(10, 6))
plt.specgram(adc_values, Fs=1/(df.index[1] - df.index[0]).total_seconds(), NFFT=256, noverlap=128, cmap='viridis')
plt.title('Spectrogram')
plt.xlabel('Time')
plt.ylabel('Frequency (Hz)')
plt.colorbar(label='Intensity (dB)')
plt.grid(True)
plt.show()
```

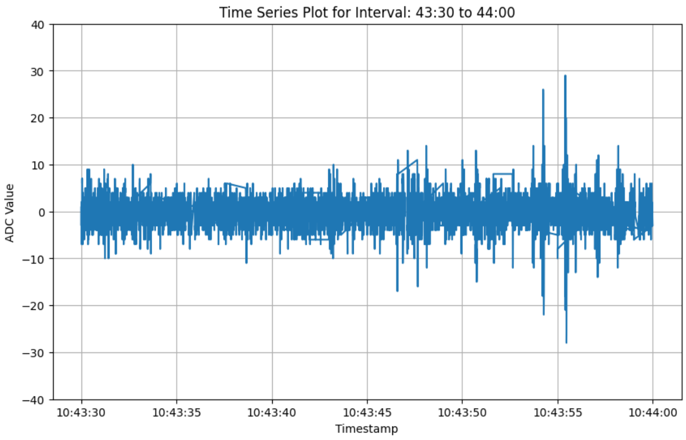
- Given Below is a breakdown of the above test in 30 seconds timeframes.

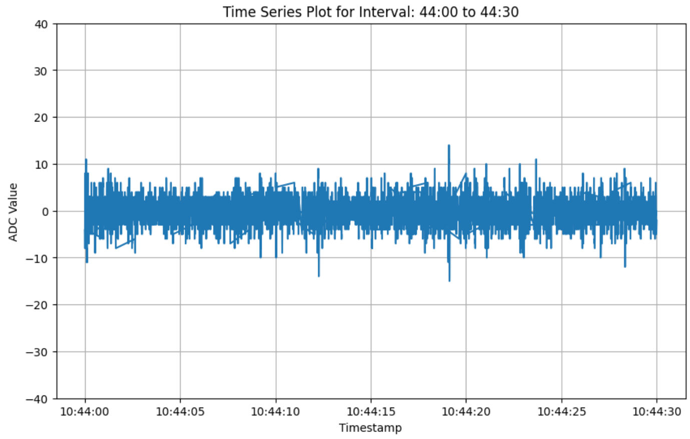


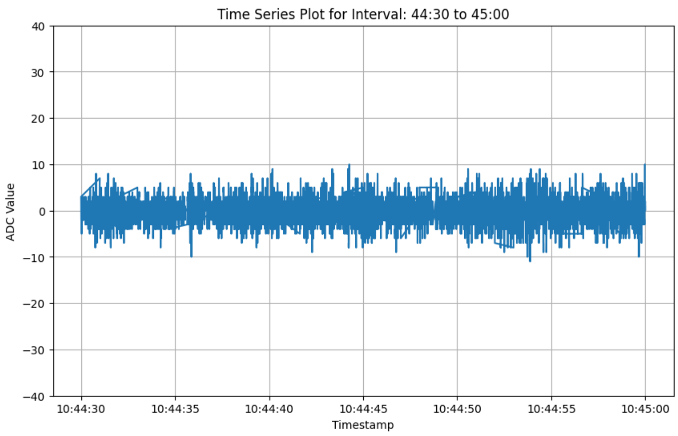


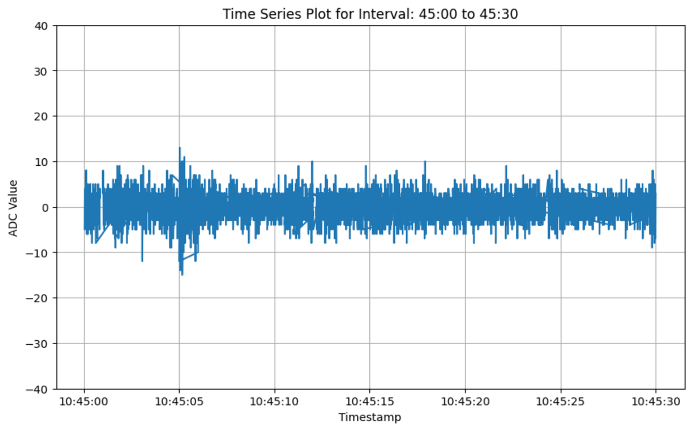

Time Series Plot for Interval: 42:30 to 43:00
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:42:30
10:42:35
10:42:40
10:42:45
10:42:50
10:42:55
10:43:00

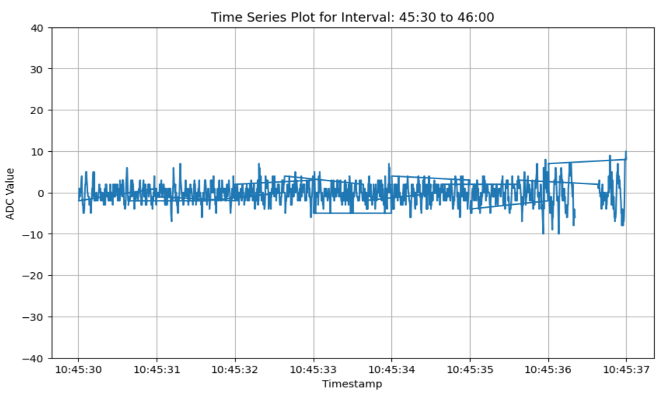

Time Series Plot for Interval: 43:00 to 43:30
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:43:00
10:43:05
10:43:10
10:43:15
10:43:20
10:43:25
10:43:30


Time Series Plot for Interval: 43:30 to 44:00
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:43:30
10:43:35
10:43:40
10:43:45
10:43:50
10:43:55
10:44:00

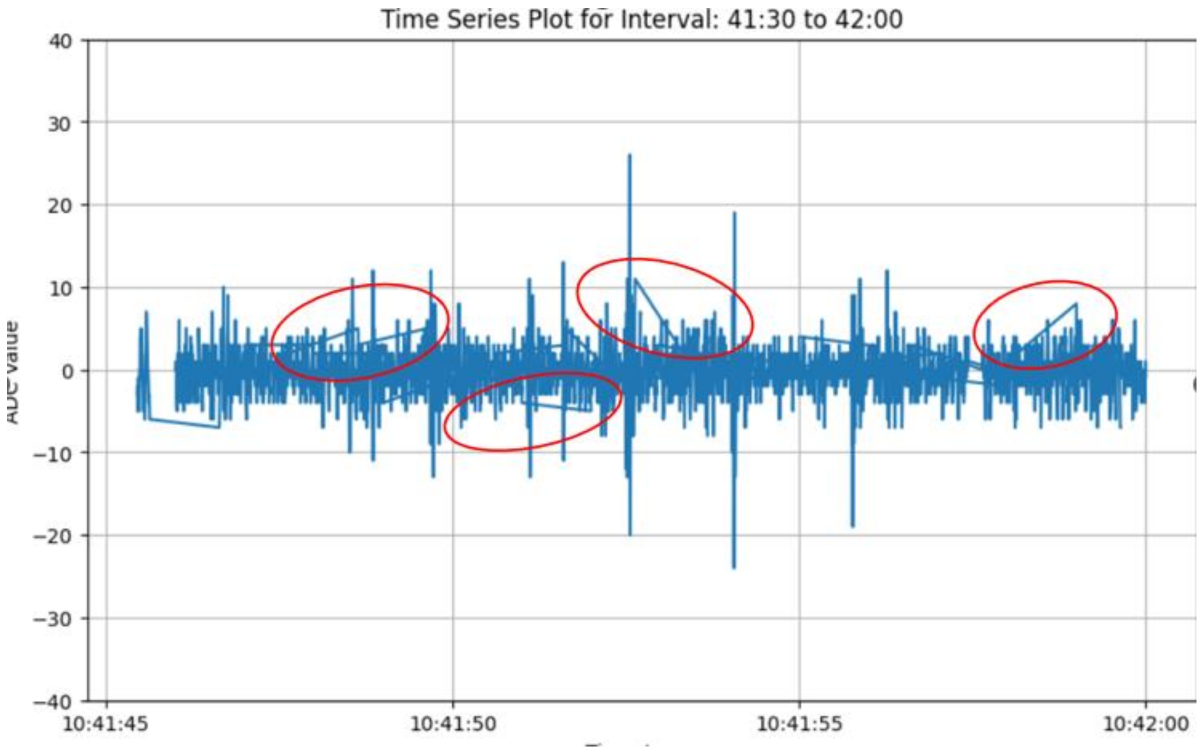

Time Series Plot for Interval: 44:00 to 44:30
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:44:00
10:44:05
10:44:10
10:44:15
10:44:20
10:44:25
10:44:30


Time Series Plot for Interval: 44:30 to 45:00
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:44:30
10:44:35
10:44:40
10:44:45
10:44:50
10:44:55
10:45:00


Time Series Plot for Interval: 45:00 to 45:30
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:45:00
10:45:05
10:45:10
10:45:15
10:45:20
10:45:25
10:45:30

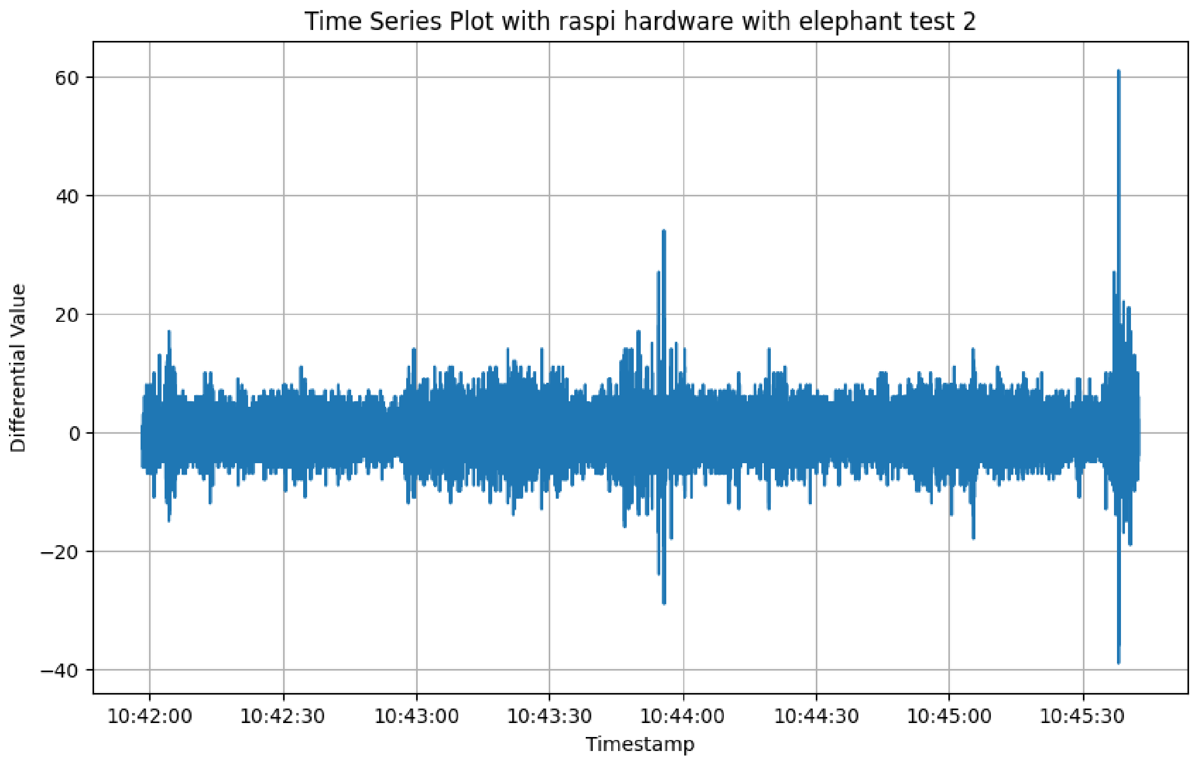

Time Series Plot for Interval: 45:30 to 46:00
ADC Value
Timestamp
40
30
20
10
0
−10
−20
−30
−40
10:45:30
10:45:31
10:45:32
10:45:33
10:45:34
10:45:35
10:45:36
10:45:37

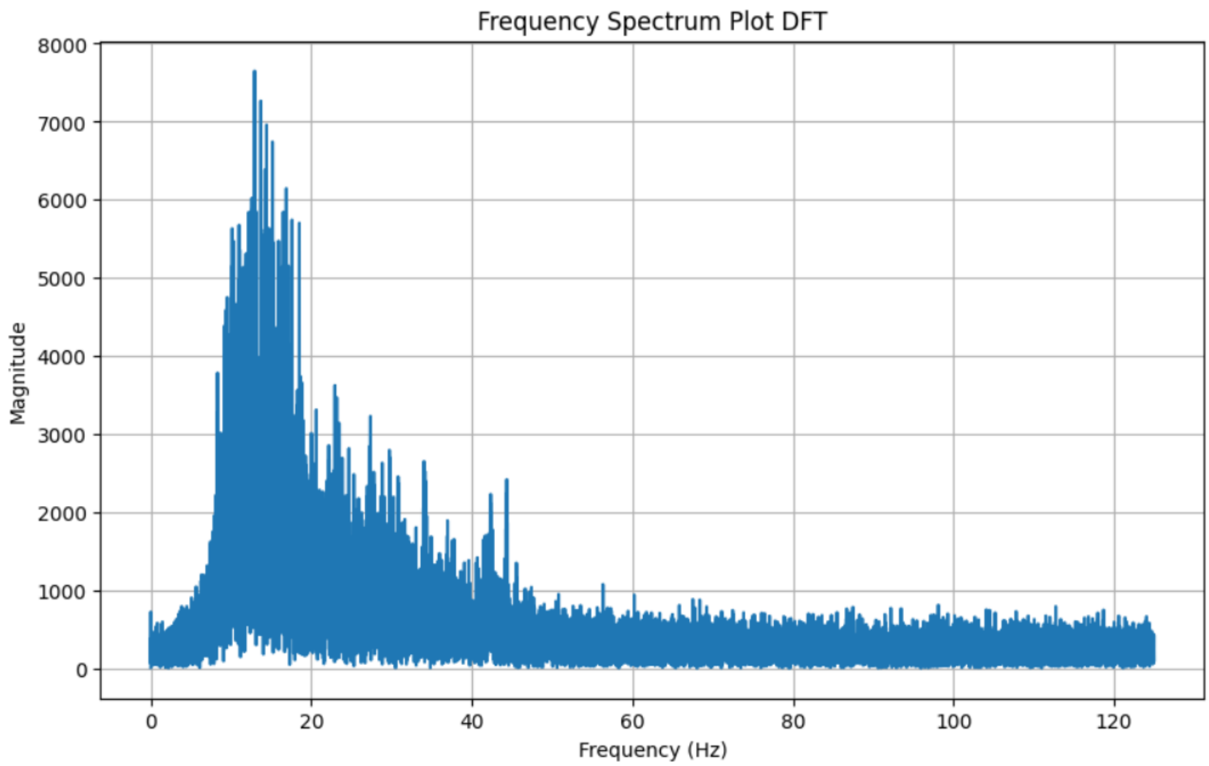
- Final time frame is the back ground noise after the test was over it was taken for few seconds after walking of the elephant was over.
- One of the problems encountered while collecting this data was that the same data was recorded in the same time frame. The errors are circled in the plot.
- Even though this seems like a Firmware error in the microcontroller the same firmware is used in the inbuilt ADC data recording mechanism.
- The Firmware developed through Arduino IDE is provided with the mail sent for further reference.

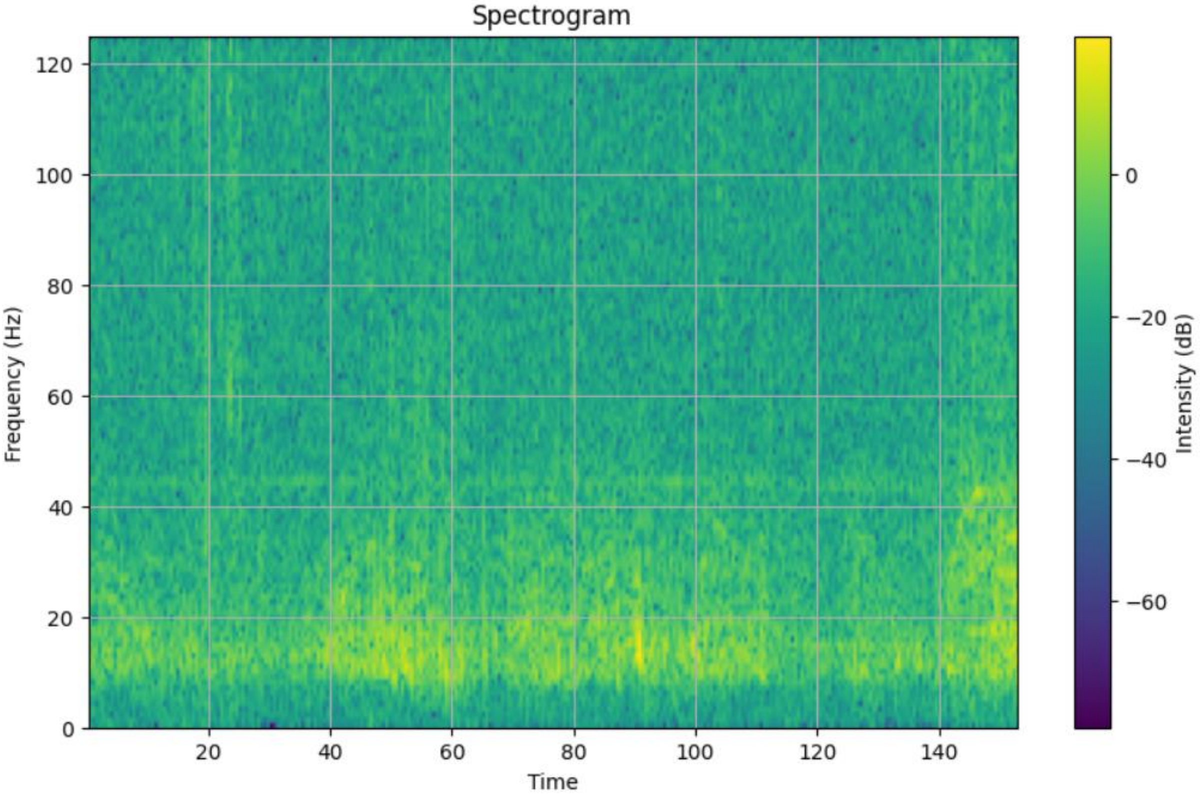


•

Data was collected in the same test using raspberry pi with the external 16-bit ADC for Cross reference purposes



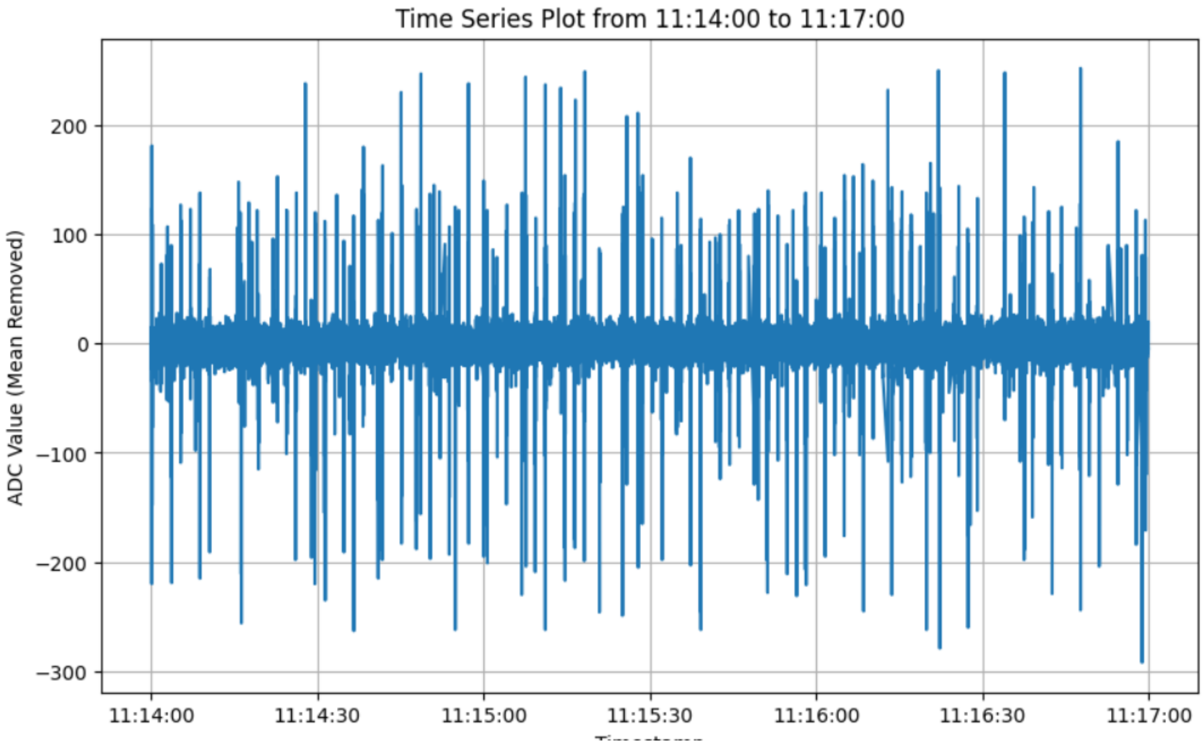


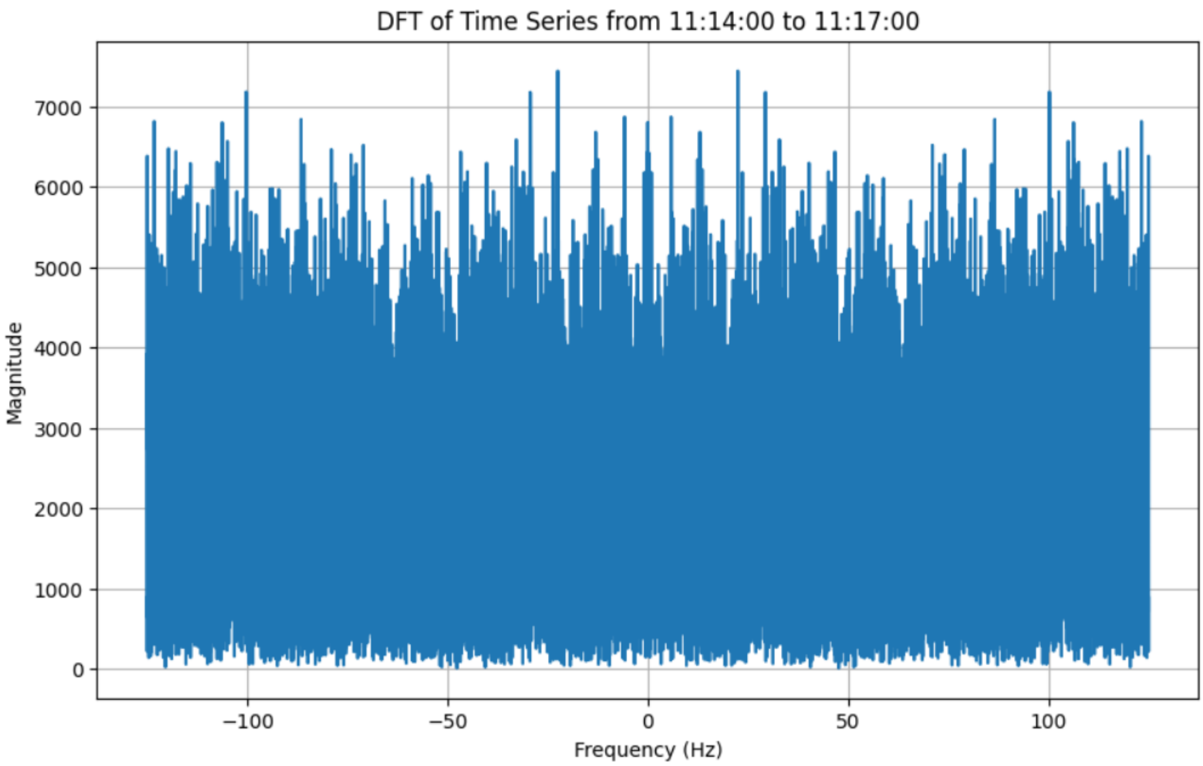


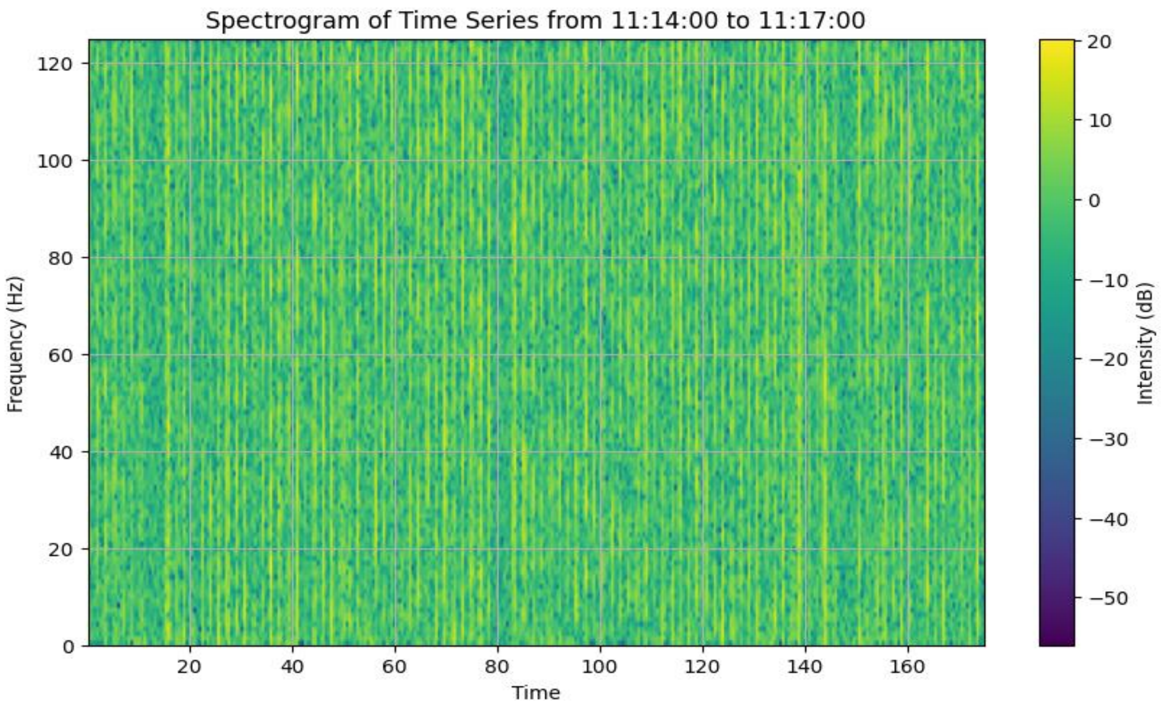
- Since raspberry pi is easy to handle and reliable we used a raspberry pi to collect the same data while using the microcontroller.
- In both of the apparatus the 16-bit external ADC was used.
- This test was just performed to cross reference the data we collected and for verification purposes
- As we see the plots give the same Data as above.
- The Codes used used to collect data in the raspberry pi is attached with the mail.

2. Using the inbuilt ADC

- Using the above methodology where walking the elephant front and backwards within a range 0f 50m the data was recorded.
- The Time series plot of the data recorded is given below.







- As we see above there isn't any dominant frequency in the dataset even after removing the DC offset value in the dataset.
- The spectrogram analysis even does not show any dominant frequency range of the elephant
- The Firmware developed for the data logging and the Python codes used for plotting is is attached with the email
- This is the problem that we are currently facing in the project which is deciding the hardware that we are going to proceed. (Whether to use a External 16 bit ADC or ESP inbuilt ADC).
- The Datasets of the tests Done up to 1/12/2024 can be accessed here.
- Further the Worked out Jupyter Notebook is attached with the mail.